

---

# Designing for Email Response Management

**Sungjoon Steve Won**

Human-Computer Interaction Institute  
School of Computer Science  
Carnegie Mellon University  
Pittsburgh, PA 15213 USA  
stevewon@gmail.com

**Laura A. Dabbish**

Human-Computer Interaction Institute  
School of Computer Science  
Carnegie Mellon University  
Pittsburgh, PA 15213 USA  
dabbish@cmu.edu

---

Copyright is held by the  
author/owner(s).

CHI 2009, April 4–9, 2009, Boston,  
Massachusetts, USA.

ACM 978-1-60558-247-4/09/04.

**Abstract**

Email is the most widely used form of computer-mediated communication. And replying to messages is one of the main activities email interfaces need to support. In this paper we address the problems users face when managing emails that need a reply. Previous work has found that users have difficulty remembering to reply to messages when they postpone response, and have trouble re-finding messages they want to respond to. We review related work on email management, and describe three designs developed to facilitate email response management.

**Keywords**

Email, computer-mediated communication, social messaging, design

**ACM Classification Keywords**

H5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

**Introduction**

Over the past thirty years, there has been a great deal of research focused on understanding email use and redesigning the email interface. However, interface

support for managing postponed responses to emails is still quite limited. Problems can occur when messages needing a response are put off to deal with at a later time and subsequently forgotten. Our goal in this work is to study how people handle emails that need a response, and to design interfaces that better support this activity. In this work-in-progress submission, we discuss some of the main issues we have identified around handling extended reply emails, and describe three designs we have developed to accommodate postponed responses.

### Problem Space

Whittaker and Sidner [6] were among the first to highlight the varied techniques users employ for managing their email messages. They found that one of the main challenges users faced with email was reading and replying to messages in a timely manner [6]. Users often put off dealing with emails that need an extended or lengthy response, and first go through potentially more urgent or manageable messages in the inbox [6]. This means the emails that are put off until later require future reference. As emails needing response pile up in a user's inbox, they may lose track of which emails need a reply [1]. This form of email overload can result in backlogs of unanswered emails, lost information, and reduced responsiveness to future email messages [6].

When users do remember that an email needs a response, they may have difficulty finding the particular message they need to reply to. Previous studies of have noted that search in email is ill-supported. Users often have trouble locating messages or remembering where they filed messages [1, 3, 5, 6].

To complement the previous work and collect more detailed data on email response management we conducted think-aloud observations with 5 users. The previous work and our think-aloud data highlighted the following issues with email response management that need to be accommodated:

1. **Recall:** Users forget to revisit emails that they put off to reply to later.
2. **Opportunistic re-finding:** Users may not notice older emails that need response because of the limited number of messages visible at the surface level in the inbox, and the fact that messages needing a reply are visually indistinguishable from other emails.
3. **Search - targeted re-finding:** Users have difficulty re-locating messages when they remember they need a response.

### Existing Techniques

In light of the challenges surrounding extended response, we examined the previous work redesigning and improving the email interface to motivate our designs. These existing techniques provided some insight into how to design for extended response.

#### *Supporting Recall*

Several design features for email allow users to explicitly mark or file email messages for later access. These kinds of features include to-do lists for email, stars, highlighting, and specialized reminder flags. These designs require the user to explicitly demarcate messages requiring future action or attention.

And when large numbers of messages are marked for re-visitation, it can become difficult to navigate the demarcated messages. Gwizdka showed the ineffectiveness of a to-do list, a calendar, general email flags, and specialized reminder flags indicating the type of action required [3].

#### *Supporting Opportunistic Re-finding*

The inbox is visited each time users check for new messages, and past research has found that because of this frequent access messages in the visible portion of the inbox serve a reminding function [1]. One way to support opportunistic re-finding is to increase the number of messages visible in the inbox. Design ideas presenting an aggregated representation of related messages, such as threading for email [5], save space in the inbox by combining multiple emails in one line.

Increasing the visual salience of certain types of messages (such as messages needing a reply) should also support opportunistically finding these messages when browsing the inbox. Many research systems for email have experimented with adding a visual weight or importance to each email. Warning bars that change as a deadline approaches [1, 3], reminders for emails [3], display of past actions taken on an email [1], and rank ordering senders by closeness have been explored for directing users' attention to important or urgent emails.

#### *Supporting Targeted Re-finding*

Locating specific messages, or targeted re-finding, is an important user behavior that supports both the communication and archiving functions of email [1, 3, 5, & 6]. One way users do this is by filing messages into folders. However, users often have trouble creating appropriate names for their folders while filing

messages away, and have difficulty remembering which folder they used when retrieving messages [6].

Recently email research has explored the utility of tagging messages with keyword meta-data to support more flexible storage and recall. Tagging emails with keywords and using a tag cloud to display the keywords makes it easier to find messages with a keyword search rather than orienteering through folders [5]. However, tagging requires significant manual input by the user. In addition, users may have the same problems with tags as with folder labels, that is generating appropriate tags and remembering what tags were used for particular messages when searching.

After considering the benefits and limitations of these existing techniques, we explored a set of new designs to support re-visitation of emails needing a response.

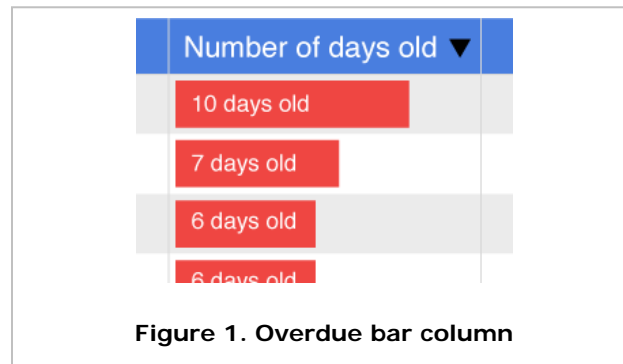
### **Redesigning the Interface**

We introduce the following three design ideas: (1) overdue bars, (2) mini-feed, and (3) message cloud. These designs are intended to support recall, targeting re-finding, and opportunistic re-finding behaviors we have identified are central to *extended reply email* (ERE) management, i.e. management of messages where reply is required but postponed to a later time.

For all three of our designs, we conceptualize that users will explicitly mark each email message as an *extended reply email* (ERE) when appropriate. We are currently exploring different interaction techniques for seamless message marking. Our designs also assume that once an ERE has been replied to, it will be automatically unmarked as an ERE.

#### *Design Idea #1: Overdue bars*

The first design idea we present is the concept of *overdue bars* to be displayed alongside each message. This idea was inspired by the task deadline indicator introduced by Bellotti et al [1]. Instead of displaying time left relative to associated task deadlines, this design shows a message specific indicator of time taken to respond. For emails that are marked by the user as ERE, our design will keep track of number of days since the email arrived in inbox. For each unique sender the user corresponds with, the system will track the user's typical response latency in days [4] and display the number of days the message is "overdue", that is how many days longer the user is taking to respond than they usually do for this person. For new senders, the system will use a ideal response threshold set by the user (e.g. reply within two days). When the user has corresponded with the new sender over at least three messages, the system will learn the users average response latency with that sender. The data per email on the number of days old feeds into the computations used in design idea #2 and design idea #3.



**Figure 1. Overdue bar column**

The overdue bar will support extended response by providing a new category for search queries and a new sortable column in the inbox (Fig. 1). By having a sortable column embedded in the inbox, the user will be able to easily see all EREs sorted in order of overdue status based on the number of days old, with the most overdue messages appearing at the top or bottom of inbox. The presence of the overdue bar indicator and the fact that users can sort and view only the subset of email messages requiring a reply should accommodate opportunistic re-finding of EREs.

#### *Design Idea #2: Mini-feed*

The second design idea we present is the *mini-feed*. The mini-feed is directly inspired by the design of Web Clips in Gmail [2]. In the Gmail interface, Web Clips are news items, RSS feeds, or sponsored-links that appear in a bar displayed above the email inbox (Fig. 2). The bar displays one item at a time and cycles through different items automatically, but the user also has the ability to navigate through the items manually with the left and right arrow navigation buttons. Clicking on the current displayed entry takes the user to the link related to the item. In the design of the mini-feed, we adapted this interaction technique to remind users about EREs. In the *mini-feed*, we insert subject lines and senders of EREs into the bar displayed above the inbox. The mini-feed will traverse through different EREs, and the user will also have the ability to navigate through them manually with the navigation buttons (Fig. 3). Clicking on the current entry would open the email in a new window. The messages are not displayed in a specific order, however, the overdue status of an ERE affects how frequently it appears in the mini-feed.

Because the mini-feed displays EREs located throughout the entire inbox, users will be reminded of EREs that they may have lost track of because they are not immediately visible. Hence, this design idea supports “opportunistic re-finding” of EREs. This design also supports “recall” by unobtrusively periodically reminding users about EREs in the inbox. In tuning this design, we still need to determine how salient the mini-feed should be to make the reminders effective but also keep them unobtrusive to users.

*Design Idea #3: Message-cloud*

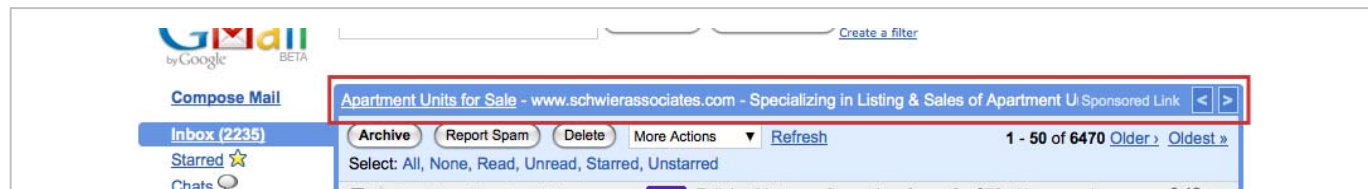
The third design idea we present is the *message-cloud*. The message-cloud representation is inspired by the design of tag-clouds [5]. In the past, tag-clouds have been used to visualize sets of items, such as web-page links, that have been “tagged” with different keywords. Visual emphasis is given to more frequently used tags, with cues such as larger font-size or different colors.

We present the idea of a “message-cloud” which uses a display quite similar to that of a tag-cloud, but requires significantly less input from the user. Instead of

different keyword tags, the message-cloud will display the subject lines and senders of EREs. In the message-cloud visual salience of a message will be determined based on the relative weight of a message as compared with the average weight of all other ERE messages (see Fig. 4). The weight of a message is calculated by combining the following attributes:

- Number of recipients in the email
- Number of times user has communicated with the sender(s): This is to predict strength of relationship between the sender(s) and the user.
- Message overdue status: This is computed by days overdue as described in design idea #1.

The message-cloud supports all three of the behaviors associated with ERE messages. The ERE information displayed in the message-cloud can act as a reminder supporting “recall”. The message-cloud representation will also support “targeted re-finding” of EREs, because the cloud representation and associated visual weighting make it easy for users to locate EREs of high



**Figure 2. Web clips in Gmail show different advertisements, news items, or RSS feeds**



**Figure 3. Mini-feed design showing subject and sender of an ERE message**

weight. Finally, because the message-cloud displays EREs located throughout the inbox, it expands the viewable area of the inbox supporting “opportunistic re-finding”. In further developing this design we must tune the weighting algorithm to appropriately balance the three attributes used. In addition, we must tune the visual design of the message-cloud display to minimize user distraction.

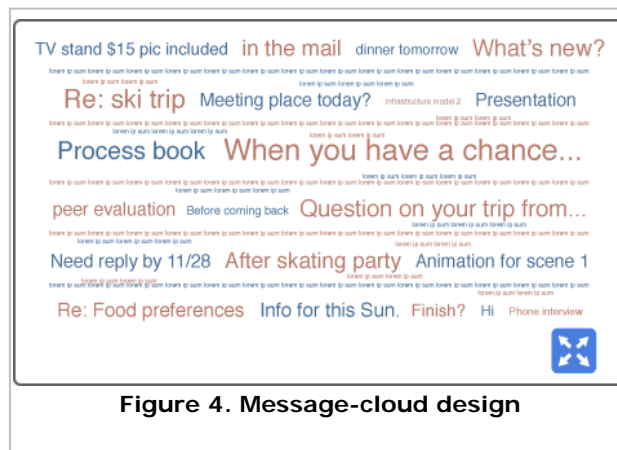


Figure 4. Message-cloud design

## Discussion and Future work

Users have long faced difficulties managing postponed email replies [6]. However, limited previous research has directly addressed the problem of managing postponed email responses. We have proposed three new design ideas that require little input from the users and support recall, opportunistic re-visitation, and targeting re-visitation of extended response emails. We are currently developing a user study to evaluate each of the proposed design ideas and compare their utility. We plan to examine the impact of each design on the

likelihood of response and speed of response to email messages that must be postponed.

## References

1. Bellotti, V., Ducheneaut, N., Howard, M., & Smith, I. (2003). Taking email to task: The design and evaluation of a task management centered email tool. *Proceedings of CHI 2003*, ACM Press: New York, 345-352.
2. GMail, "About Web Clips", <http://mail.google.com/support/bin/answer.py?hl=en&ctx=mail&answer=18219>, verified December 20, 2008.
3. Gwizdka, J. (2002). TaskView: Design and evaluation of a task-based email interface. In *Proceedings of CASCON'2002*, Toronto, CA. 136-145.
4. Kalman, Y.M., Ravid G., Raban, D.R. & Rafaeli S. (2006). Pauses and Response Latencies: A Chronemic Analysis of Asynchronous CMC. *Journal of Computer Mediated Communication*, 12(1), 1-23.
5. Tang, J., Wilcox, E., Cerruti, J., Badenes, H., Nusser, S., Schoudt, J. (2008). Tag-it, snag-it, or bag-it: Combining tags, threads, and folders in e-mail. In *Extended Abstracts of CHI 2008*, ACM Press: New York, 2179-2194.
6. Whittaker, S., & Sidner, C. (1996). Email overload: Exploring personal information management of email. *Proceedings of the ACM Conference on Human Factors in Computing Systems*. ACM Press: New York, 276-283.